REMARKS

This Amendment and Response is intended to fully respond to the Final Office Action mailed February 2, 2006. In that Office Action, claims 1, 2, 4, 6-22, 24, 26, 27 and 29-33 were examined, and all claims were rejected. More specifically, claims 1, 2, 6, 9, 11-20, 24, 26, 27, 29 and 31 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Burridge (USPN 6,918,106), hereinafter "Burridge," in view of "Enterprise JavaBeans" by Monson-Haefel, hereinafter "Haefel;" claims 4 and 10 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Burridge in view of Haefel as applied to claim 1, and further in view of "Method to Update Java Class Library in Client Computer at Runtime" by IBM Technical Disclosure, hereinafter "IBM;" claims 7, 8, 30, 32 and 33 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Burridge in view of Haefel as applied to claims 1 and 26, and further in view of Chan (USPN 6,470,494) , hereinafter "Chan;" and claims 21 and 22 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Burridge in view of Haefel as applied to claim 1, and further in view of Menachemi (US Pub. No. 2002/0103810 A1) , hereinafter "Menachemi." Reconsideration of these rejections, as they might apply to the original and amended claims in view of these remarks, is respectfully requested.

In this Response, claims 1, 24 and 26 have been amended and no new claims have been added or cancelled.

Claim Rejections – 35 U.S.C. § 103

Claims 1, 2, 6, 9, 11-20, 24, 26, 27, 29 and 31 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Burridge in view of Haefel. Applicants respectfully traverse the section 103 rejections for all claims. The claims as amended preclude a finding of a prima facie case of obviousness because one or more of the requirements of a prima facie case are absent. Indeed, a prima facie case can only be met when **all** of the following requirements are met: (1) the combined references must teach or suggest all the claim limitations; (2) there must be some suggestion or motivation in the references themselves (or in the knowledge available to those skilled in the art) to combine the references; and (3) there must be a reasonable expectation of success. See MPEP §§ 706.02(j) and 2143. In this case, neither Burridge nor Haefel teach or disclose generating the customized library, including *only* the one or more client-needed types,

where the client-needed types comprise one or more of the application-referenced types that are not client-loaded types.

The present invention includes a customized library management method and system for generating a customized library needed for execution of an application in a client system. In response to an identification of a given application, such as a request from the client system or an internal instruction of the server, the server determines the appropriate types to include in a library to be sent to the client based on certain parameters. The parameters may include, for example, the types referenced by the application; the types already "loaded" on the client system, and a device profile describing characteristics of the client system. The customized library includes only a subset of the types, referred to in the exemplary embodiments as the client-needed types, that are required by the application; the client-needed types include all application-referenced types that are not already "loaded" on the client. A type is loaded on a client if the type is present, resides, or is stored at the client. As such, only the client-needed types are included in the customized library, i.e., only those application-referenced types not already on the client.

Burridge describes methods for optimizing a program during its runtime execution. See col. 2, 45-51 ("a method and apparatus for determining which program units are loaded *during execution* of a dynamically loaded program"). More particularly, Burridge creates a library file of program files "loaded" during execution of a main program unit. See col. 2, lines 56-64 ("creating at least one library file containing only application program files loaded *during the first execution*"). Burridge describes loading as loading a program unit at run time before the unit is used, similar to invoking an object before executing a contained method in the object. See col. 2, lines 38-39 ("Each reference program unit must be loaded at run time before the referenced unit is used").

To create the library, Burridge describes an offline loader that executes a main program unit. See col. 5, lines 16-18. The offline loader waits for the main program to request a program unit or file, similar to an object requesting another object, and loads the program unit from the client source. See col. 5, lines 19-22 ("If the offline loader 26 requires a program unit that has not already been loaded, the offline loader 26 obtains the program unit from the pathnames

specified.") The program unit already exists on the client or can be accessed by the main program already on the client. The offline loader then writes a copy of the program unit to a library. See col. 5, lines 22-25 ("the offline loader 26 writes a copy of the program file to an application library file"). The library file is then used for any subsequent executions of the main program. See col. 5, line 66 – col. 6, line 1 ("The library file created at reference numeral 38 is used as the input source for application program files in subsequent executions of the main program unit."). At a later execution of the main program, the runtime loader obtains the program units from the library file, which reduces the overhead in obtaining application programs from multiple sources. See col. 6, lines 5-9.

Haefel describes Java archives (JAR files). More particularly, Haefel describes deployment descriptors that can be used with JAR files to customize the behavior of software at runtime. See page 30, paragraphs 4-6.

Independent Claim 1, Claim 24, and Claim 26

Amended claims 1, 24, and 26 are allowable over Burridge and Haefel either in combination or alone. Neither Burridge nor Haefel teach or disclose generating the customized library, including *only* the one or more client-needed types, where the client-needed types comprise one or more of the application-referenced types that are not client-loaded types. Burridge, the primary prior art reference, and the present invention as defined in the claims have several fundamental differences, but, Applicants wish to focus on the difference explained below.

Burridge does not teach generating the customized library, including *only* the one or more client-needed types, where the client-needed types comprise one or more of the application-referenced types that are not client-loaded types. The library file described in Burridge, "contains all application program files 22 loaded during the execution of the main method." Col. 5, lines 26-28. To create the library file, the offline loader executes the main method and copies every file, as it is loaded, into the library file. See col. 5, lines 18-22. The offline loader simply loads every file needed by the main method into the library file. There is no mention in Burridge of identifying client-needed types, which are any application-referenced types not already on the

client. The library in Burridge essentially contains all application-referenced types, whether a client-loaded type or client-needed type.

The customized library, of the present invention, is different because the customized library includes only client-needed types. The customized library, thus, has fewer than all the application-referenced types, as opposed to the library described in Burridge, because the customized library excludes any type already stored or present on the client.

Further, the library described in Burridge includes application types that are "loaded," applying the term "loaded" as explained in present invention. Burridge includes, in the library, types that are present on the client and used during the execution of the main program. See col. 4, lines 39-41 ("application program files required for execution of a dynamically loaded program are collocated in a library file.") and generally, col. 2, lines 20-25 (showing that if files are not found in local JAR files, then the a browser may search the web with a HTTP request). Therefore, the library taught in Burridge includes client-loaded types, which are explicitly excluded from the customized library claimed in the present invention. Thus, the present invention as defined in the claims provides a marked improvement of Burridge because the smaller library file will take less time to send to the client.

Haefel does not overcome the shortcomings of Burridge. Haefel describes deployment descriptors that can be used with JAR files to customize the behavior of software at runtime. See page 30, paragraphs 4-6. There is also no mention in Haefel about generating the customized library, including *only* the one or more client-needed types, where the client-needed types comprise one or more of the application-referenced types that are not client-loaded types. Therefore, Haefel does not teach this element of the claims.

For the reasons stated above, independent claims 1, 24, and 26 are allowable over the prior art. Likewise, claims 2-4, 6-22, 27, 29-33 depend from the allowable independent claims and thus, are also allowable. Examiner is respectfully requested to remove the rejections to the amended claims and allow all claims now present.

Conclusion

This Amendment fully responds to the Final Office Action mailed on February 2, 2006. It is recognized that the Office Action may contain arguments and rejections that are not directly addressed by this Amendment due to the fact that they are rendered moot in light of the preceding arguments and amendments in favor of patentability. Hence, the failure, if any, of this Amendment to directly address an argument raised by the Examiner should not be interpreted as reflecting the Applicants' belief that such argument has merit. Furthermore, the claims of the present application may include other elements, not discussed in this Amendment, which are not shown, taught, or otherwise suggested by the art of record. Accordingly, the preceding arguments in favor of patentability are advanced without prejudice to other bases of patentability.

It is believed that no further fees are due with this Response. However, the Commissioner is hereby authorized to charge any deficiencies or credit any overpayment with respect to this patent application to deposit account number 13-2725.

In light of the above amendments and remarks, it is believed that the application is now in condition for allowance, and such action is respectfully requested. If the Examiner believes a telephone conference would advance the prosecution of this application, the Examiner is invited to telephone the undersigned at the below-listed telephone number.

Respectfully submitted,

Date: 4/3/06

Tadd F. Wilson
Reg. No. 54,544
MERCHANT & GOULD P.C.
P.O. Box 2903
Minneapolis, Minnesota 55402-0903
303.357.1651